



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06F 15/18, 15/80, 7/02	A1	(11) International Publication Number: WO 91/19259 (43) International Publication Date: 12 December 1991 (12.12.91)
--	----	--

(21) International Application Number: PCT/US90/03068

(22) International Filing Date: 30 May 1990 (30.05.90)

(71) Applicant (for all designated States except US): ADAPTIVE SOLUTIONS, INC. [US/US]; 1400 NW Compton Drive, Suite 340, Beaverton, OR 97006 (US).

(72) Inventor; and

(75) Inventor/Applicant (for US only): HAMMERSTROM, Daniel, W. [US/US]; 19973 SW Oak Court, Aloha, OR 97707 (US).

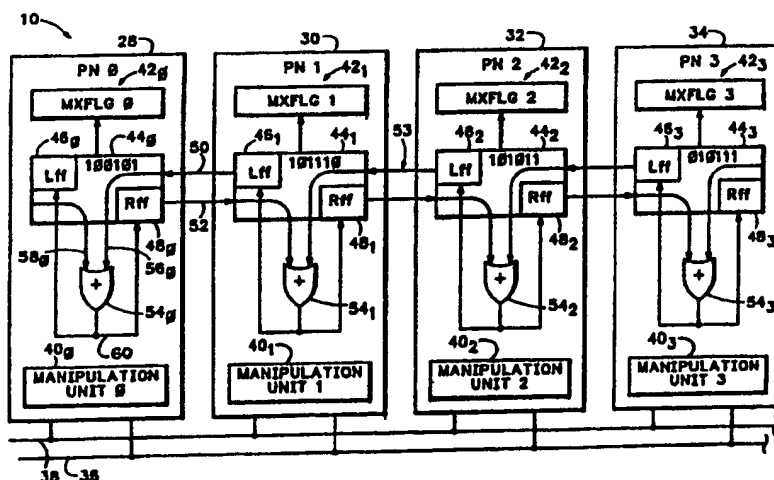
(74) Agents: VARITZ, Robert, D. et al.; 200 Pacific Building, 520 SW Yamhill Street, Portland, OR 97204 (US).

(81) Designated States: AT (European patent), BE (European patent), CA, CH (European patent), DE (European patent)*, DK (European patent), ES (European patent), FR (European patent), GB (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent), US.

Published

With international search report.

(54) Title: DISTRIBUTIVE, DIGITAL MAXIMIZATION FUNCTION ARCHITECTURE AND METHOD



(57) Abstract

A maximization architecture includes an array (10) of processor nodes (28, 30, 32, 34) wherein each node has a manipulation unit (40) contained therein. Each node is connected to an input bus (36) and to an output bus (38). A data register (44) is located in each node and contains a data figure, which consists of plurality of segments, or bits, wherein each segment or bit has a value. A maximization mechanism (42) is located in each node and is connected to an arbitration bus (53) which extends between adjacent nodes and an arbitration mechanism (54, 46, 48), which is connected to the arbitration bus (53), for comparing a value of a bit to a signal which is transmitted on the bus and for subsequently transmitting a comparison indicator. Each node includes first (46) and second (48) retainers, which are connected to the arbitration means for retaining the comparison indicator. A flag mechanism (62) is provided and flags the processor node which contains the maximum data figure.

DESIGNATIONS OF "DE"

Until further notice, any designation of "DE" in any international application whose international filing date is prior to October 3, 1990, shall have effect in the territory of the Federal Republic of Germany with the exception of the territory of the former German Democratic Republic.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LJ	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark				

- 1 -

DISTRIBUTIVE, DIGITAL MAXIMIZATION FUNCTION ARCHITECTURE AND METHOD

Technical Field

The instant invention relates to a computer processor architecture,
5 and specifically to an architecture which provides a maximization structure and method of determining which of several processor nodes in an array contains a maximum data value.

Background Art

Neural networks are a form of architecture which enables a
10 computer to closely approximate human thought processes. One form of neural network architecture enables single instruction stream, multiple data stream (SIMD) operations which allow a single command to direct a number of processors, and hence data sets, simultaneously.

There are several important practical problems that cannot be
15 solved using existing, conventional algorithms executed by traditional, conventional computers. These problems are often incompletely specified and are characterized by many weak constraints requiring large search spaces.

The processing of primary cognitive information by computers, such as computer speech recognition, computer vision, and robotic control, fall
20 into this category. Traditional computational models bog down to the point of failure under the computational load if they are tasked to solve these types of problems. Yet animals perform these tasks using neurons that are millions of times slower than transistors. Feldman's 100-step rule states that a "human" cognitive process having a time of 500 msec. can be accomplished in 5 msec
25 neuron switching time. This implies that there are two vastly different computational models at work. It also suggests that in order to build computers that will do what nervous systems do, the computers should be structured more like nervous systems.

A nervous system, and a neurocomputational computer, is
30 characterized by continuous, non-symbolic, and massively parallel structure that is fault-tolerant of input noise and hardware failure. Representations, ie, the input, are distributed among groups of computing elements, which independently reach a result or conclusion, and which then generalize and

- 2 -

interpolate information to reach a final output conclusion.

Put another way, connectionist/neural networks search for "good" solutions using massively parallel computations of many small computing elements. The model is one of parallel hypothesis generation and relaxation to the dominant, or "most-likely" hypothesis. The search speed is more or less independent of the size of the search space. Learning is a process of incrementally changing the connection (synaptic) strengths, as opposed to allocating data structures. "Programming" in such a neural network is by example.

One particularly useful operation which is performed by neural networks is character recognition, which may be used to input printed or handwritten material into an electronic storage device. It is necessary for the system to recognize a particular character from among hundreds of possible characters. A matrix may be provided for each possible character, which matrix is stored in a processor node, for comparison to a similar matrix which is generated by analyzing the character to be stored. The input matrix is compared, by all processor nodes to the values contained therein, and the best match, the maximum correlation between the stored data and the input data, determines how the input matrix will be interpreted and stored electronically.

The problem of how to distribute the comparison function and determination of best match across a processor array with potentially thousands of processor nodes may be solved in a variety of ways. The most efficient way to determine a best match would be to use an analog system, where the magnitude of a current is proportional to the number being maximized. There are a number of electrical problems with this approach. The analog approach has limited precision and will not work well across multiple integrated circuits, ie, it is usually limited to a single chip and therefor limited to the number of PNs which may be placed on a single chip.

Disclosure of the Invention

An object of the invention is to provide a maximization architecture which determines, in an array of processor nodes, which processor node has the maximum data figure contained therein.

Another object of the invention is to provide a maximization

- 3 -

architecture which allows selectable, arbitrarily large, precision of determining maximization.

5 A further object of the invention is to provide the maximization architecture which analyzes a data value on a bit-by-bit method to produce a winner-take-all result without necessarily analyzing ever bit of every data figure contained in the array of processor nodes.

Another object of the invention is to provide a method of determining which of an array of processor nodes contains a maximum data value.

10 Still another object of the invention is to provide a method of determining, without necessarily examining every bit of data value which is the maximum data value contained in an array of processor nodes.

Yet another object of the invention is to provide a method of determining, without necessarily examining every bit of data value which is the maximum data value contained in an array of processor nodes which extend
15 over multiple integrated circuits.

The maximization architecture of the invention includes an array of processor nodes wherein each node has a manipulation unit contained therein. Each node is connected to an input bus and to an output bus. A
20 data register is located in each processor node and contains a data figure, which consists of a plurality of segments, or bits, wherein each segment or bit has a value. A maximization mechanism is located in each processor node and is connected to an arbitration bus which extends between adjacent processor nodes and an arbitration mechanism, which is connected to the arbitration bus,
25 for comparing a value of a bit to a signal which is transmitted on the arbitration bus and for subsequently transmitting a comparison indicator. Each processor node includes first and second indicator retainers, which are connected to the arbitration means for retaining the comparison indicator. A flag mechanism is provided and flags the processor node which contains the
30 maximum data figure.

The method of the invention includes providing the previously identified structure and initially setting the flag mechanism of each processor node with a positive flag. The value of a subject bit is examined in the data

- 4 -

registered to determine whether it has a value of one or zero. If and only if the value is zero, and the value of the subject segment of at least one other processor node is one, the flag register of a designated processor node is set with a negative flag. The value of the flag register is placed into the arbitration mechanism and transmitted to the indicator retainers. The steps are reiterated until only one processor node has a positive flag. The remaining positive flag indicates that particular processor node contains the maximum data figure.

These and other objects and advantages of the invention will become more fully apparent as the disclosure which follows is read in conjunction with the drawings.

Brief Description of the Drawings

Fig. 1 is a schematic diagram of a broadcast communication pattern of communication nodes contained within processor nodes of the invention.

Fig. 2 is a block diagram of a portion of an array of processor nodes of the invention containing the maximization architecture of the invention.

Fig. 3 is a block diagram of a single processor node of the invention.

Fig. 4 is a block diagram of the maximization function of the invention.

Best Mode for Carrying Out the Invention

Referring initially to Fig. 1, broadcast patterns in an array 10 of processor nodes (PNs) which contain connection nodes (CNs) 0-7 (12, 14, 16, 18, 20, 22, 24 and 26, respectively) are depicted. A CN is a state associated with an emulated node in a neural network located in a PN. Each PN may have several CNs located therein. The CNs are often arranged in "layers", with CN0 - CN3 comprising one layer, while CN4 - CN7 comprise a second layer. The array depicted would generally include four PNs, such as PN0, PN1, PN2 and PN3, (28, 30, 32 and 34, respectively) depicted in Fig. 2, with CN0 and CN4 being located in PN0, CN2 and CN5 being located in PN2, etc. There may be more than two layers of connection nodes in any one processor node

- 5 -

or in any array of processor nodes. A typical array of processor nodes may include hundreds or thousands of individual PNs.

The connection nodes operate in what is referred to as a broadcast hierarchy, wherein each of connection nodes 0-3 broadcast to each of connection nodes 4-7. An illustrative technique for arranging such a broadcast hierarchy is disclosed in U.S. Patent No. 4,796,199, NEURAL-MODEL INFORMATION-HANDLING ARCHITECTURE AND METHOD, to Hammerstrom et al, January 3, 1989, which is incorporated herein by reference.

Conceptually, the available processor nodes may be thought of as a "layer" of processors, each executing its function (multiply, accumulate, and increment weight index) for each input, on each clock, wherein one processor node broadcasts its output to all other processor nodes. By using the output processor node arrangement described herein, it is possible to provide n^2 connections in n clocks using only a two layer arrangement. Known, conventional SIMD structures may accomplish n^2 connections in n clocks, but require a three layer configuration, or 50% more structure. The boundaries of the individual chips do not interrupt broadcast through processor node arrays, as the arrays may span as many chips as are provided in the architecture.

All of the PN in array 10 are connected to an input bus 36 and an output bus 38. Each processor node includes a manipulation unit 40, which is depicted in greater detail in Fig. 3 and will be described later herein.

The maximization architecture of each PN is depicted generally at 42. The maximization architecture includes a data register 44 which contains a data figure. Subscripts are used in connection with the reference number to designate the particular PN and which a structure is located. For instance, each PN includes a data register 44. Reference numeral 28 refers to PN₀, which includes data register 44₀ therein. The data figure consists of a plurality of segments or bits, each segment or bit having a value. The technique which is used to arrive at the value is data register 44 will be described subsequently herein. Each PN also includes a left flip-flop 46 and a right flip-flop 48. The left flip-flops are also referred to herein as a first

- 6 -

indicator retainer means while the right flip-flops are referred to herein as a second indicator retainer means. As depicted in Fig. 2, each left flip-flop is connected to the data register in the left most, immediate-adjacent processor node by a connection 50. Likewise, the right flip-flops are connected to the
5 immediate right most, immediate-adjacent processor node data register by a connection 52. Connections 50 and 52 comprise what is referred to herein as an arbitration bus 53. It should be appreciated that while the connections are shown extending directly between the appropriate flip-flops and the data register, the arbitration bus may be formed as part of the input/output bus
10 structure.

Each processor node includes an OR gate 54, which is also referred to herein as arbitration means. The inputs to OR gates 54 are designated by reference numerals 56, 58 and come from connections 50, 52 respectively. The outputs 60 of OR gates 54 are connected to the left and
15 right flip-flops.

A max flag register 62, also referred to herein as flag means or maximum value indicator, receives and holds a value based on the comparison and arbitration which takes place amongst the maximization architecture of the various processor nodes.

Turning now to Fig. 3, the components of manipulation unit 40 will be described in greater detail. Manipulation unit 40 includes an input unit 62 which is connected to input bus 36 and output bus 38. A processor node controller 64 is provided to establish operational parameters for each processor node. An addition unit 66 provides for addition operations and receives input
20 from input unit 62. A multiplier unit 68 is provided and is connected to both the input and output buses and addition unit 66.

A register unit 70 contains an array of registers, which may include data register 44 as well as flag register 62. In the preferred embodiment, each processor node includes an array of 32 16-bit registers. A
25 number of other arrangements may be utilized.

A weight address generation unit 72 is provided and computes the next address for a weight memory unit 74. In the preferred embodiment, the address may be set in one of two ways: (1) by a direct write to a weight
30

- 7 -

address register or (2) by asserting a command which causes the contents of a weight offset register to be added to the current contents of the memory address register, thereby producing a new register.

5 An output unit 76 is provided to store data prior to the data being transmitted on output bus 38. Output unit 76 may include an output buffer, which receives data from the remainder of the output unit prior to the data being transmitted on the output bus. The data is transmitted to output bus 38 by means of one or more connection nodes, such as CN0 or CN4, for instance, which are part of the output unit of PN0.

10 While only single input and output buses are depicted in the drawings for the sake of simplicity, it should be appreciated that multiple input and/or output buses may be provided and if so, the various components of the manipulation unit will have connections to each input and output bus. Additionally, the input/output bus may be arranged to connect only to the
15 input and output units, and a separate internal PN bus may be provided to handle communications among the various components of the PN.

Returning now to Fig. 2, the operation of processor node array
10 as it determines which processor node has a maximum data figure contained therein will be described. This description begins with the assumption that
20 each PN in the array has a data figure therein, which is the result of the manipulation of data by the PN and that the maximization function of the invention will determine which PN in the array has the maximum data figure contained in its data register.

Turning now to Fig. 4, the maximization function of the invention
25 is depicted generally at 78. The first step in the maximization function is to set max flag (mxflg) 62 to "1" in each PN, block 80. Next, flip-flops 46 and 48, also referred to as flag registers, are cleared, i.e., set to zero, block 82.

At this point, the most significant bit (MSB), i.e., the left most bit, in data registers 44 is examined, block 84. Block 86 asks whether the most
30 significant bit is equal to one. If the answer to 86 is "yes", the value of mxflg is loaded into OR gate 54, which results in a "1" being propagated throughout the right and left flip-flops to all of the processor nodes in the array, block 88. This step enables any processor node that answered "no" to block 86 to inquire

- 8 -

whether any other most significant bit was equal to one, block 90. If block 90 is answered in the affirmative, the mxflg is set to zero for that designated processor node, block 92.

5 If the answer to block 90 is "no", the branches of the block diagram rejoin, and the array determines if a counter is greater than the number of bits in the data register, block 94. If the answer is "no", the data register is shifted left, block 96, the counter is incremented, block 98, and the routine returns to block 82 where the flag registers are cleared.

10 The function is iterated until such time as only one mxflg is equal to "1", or the counter is greater than the number of bits originally located in data registers 44. A "yes" answer to block 94 indicates that there are at least two PNs which have equal values in them, all of the bits in data register 44 having been compared and two or more of the registers containing equal values. In this situation, a tie-breaker routine may be executed, block 100.
15 Tie-breaker routine 100 may be determined by individual programmers who decide, according to the criteria being measured, which PN will be determined to hold the value of interest, and consequently, the data of interest.

As previously noted, electrical delays preclude the use of a single wire or bus to connect all of the PNs. The use of a single wire or bus is
20 simply not sufficient to overcome the electrical problems. Neither can a single wire or bus be used to cross multiple chips.

It is possible to connect all of the processor nodes together on a single line or bus, precharge the line or bus in a single phase with the input matrix, examine all of the processor nodes for the best match, and then
25 discharge the line or bus on the next phase. This solution may be operable with a few processor nodes, but with hundreds or thousands of processor nodes, the physical organization required becomes unwieldy. The first difficulty is that it will take more than one phase, or clock, for a single processor node to discharge the line. The second difficulty is that, as other functions are carried
30 out by the processor nodes, internal counters and indicators may change values. Finally, the required circuitry becomes too large to quickly conduct the signals synchronously.

The motivation for providing a right and left flip-flop in each PN

- 9 -

is to overcome the electrical problems associated with a single connection. Each flip-flop acts similarly to an axon in an animal neuron: it is set by the first signal, and the signal "remains" in the flip-flop until the flip-flop is instructed to change the signal. Therefor, when one PN propagates a "1", the
5 "1" is propagated indefinitely to all other PNs in the network until the mechanism is cleared.

To briefly follow the maximization function through a determination of which of PNs 0-3 have the largest data figure, in the first iteration, mxflg3 42₃, is set to zero, a negative flag, because the MSB in data
10 register 44₃ is zero, while the MSBs of the data registers in PNs 0-2 is "1". OR gates 54₀, 54₁, and 54₂ will all receive a "1" from their respective flag registers, and a signal which is representative of "1" will be propagated to all of the left and right flip-flops in all of the PNs in array 10. Because the counter is still low, block 94, ie, the routine has been iterated fewer times than
15 there are bits in the data register, the segments of the data registers will be shifted, block 96, the counter incremented, block 98, and the max function iterated again beginning with block 82, clearing the flip-flops.

In the second iteration, all of the remaining MSBs are 0, so no mxflgs are changed. It should be remembered that while PN3 is no longer
20 participating in the examination step, block 84, it is still propagating right and left through the flip-flops. At the end of the second iteration, the data registers will shift left again and the third iteration will begin.

In the third iteration, PN0 will set its mxflg to zero. PN2 will set its mxflg to zero in the fourth iteration, which will result in the end of the max
25 function. The function will be iterated until block 94 is answered "yes". If there is no tie, as in the case being described, the tie breaker routine, block 100, if present, will ignore the data, and the maximization function will be ended, block 102. Thus, PN2 will be determined to have the largest data figure by the maximization function and architecture of the invention. This
30 means that PN2 contains the best match to the input data and represents the particular matrix value being sought.

The following code is a simplification of the code that describes the actual CMOS implementation of the PN structure in a neurocomputer chip.

- 10 -

The code shown below is in the C programming language embellished by certain predefined macros. The code is used as a register transfer level description language in the actual implementation of the circuitry described here. Bolded text indicates a signal, hardware or firmware component, or a phase or clock cycle.

The **ph1** and **ph2** variables simulate the two phases in the two-phase, non-overlapping clock used to implement dynamic MOS devices.

The post-fix "**_D**" on each signal name, means a delayed version of the signal, "**_B**" means a bus (more than one signal line), "**_1**" means a dynamic signal that is only valid during **ph1**. These can be combined arbitrarily. Maximization arbitration logic, maximization function test signal (**rmxrd**), and the value in the **mxflg** register 42 must be set explicitly by the programmer before beginning the maximization function.

```
maxin = leftff OR rightff;
```

leftff and **rightff** are the flip-flops which are set by signals, **frmleft**, **frmright**, arriving from the PNs directly to the left and to the right of the specific PN. **maxin** is then sampled and the values on the arbitration bus (50, 52) tested:

```
if ( (ph2) ANDb (acval) ANDb (rmxrd_2) ) {
    if (mxflg) {
        if ( (logneg==0) ANDb (maxin==1) ) mxflg=0;
    }
}
```

acval indicates that the max function test signal, **rmxrd_2**, is valid. **logneg** is the most significant bit of the logic/shifter output. The shifter is used to shift the value being checked left by one bit each iteration. **mxflg** indicates whether this PN is still in the running for the maximum value. If **maxin** is 1 and **logneg** is zero, then it knows that there is at least one other PN that has a larger number (a 1 in the same position that he has a 0).

```
if (((ph1) ORb (ph2)) ANDb (acval) ANDb (clrdn))
{
    rightff=0; leftff=0; toleft_B=0; toright_B=0;
    frmleft_B=0; frmright_B=0; }
}
```

clrdn clears flip-flops 46, 48 which indicate the state of other PNs in the array to the PN in which flip-flops 46, 48 are located. Signals are provided to the PNs to the left and right of the specific PN. The **rightff** and

- 11 -

leftff signals are used to emulate a large OR gate and cooperate with OR gates 54. That is, they are cleared and then whenever any PN generates a signal, which is done whenever data register 44 has a 1 as the most significant bit of the data figure being maximized (and shifted) the flip-flops are set by the PN with the 1 which propagates a signal along the arbitration bus. The architecture also propagates signals left and right, like dominos. The result is that if any PN drives its flip-flops, all flip-flops in all PNs are set. Each PN can read its own flip-flops to determine if anybody drove the "OR" function, which indicates that some PN has a 1 as the most significant bit in its data register.

10 The following is a description of the structure and signals which are used to propagate values from the OR gates in both directions along the arbitration bus.

```

15           if ((mxen) ANDb (glbarb==0) ANDb (segarb==0)) {
           rghtff OR= BIT(frmrght_B,0);
           leftff OR= (BIT(frmleft_B,0) AND INV(mxsw));
           leftff OR= rghtff;
           rghtff OR= leftff;
20           if ((ph1) ANDb (acval) ANDb (rmxdrv_1) ANDb
(logneg) ANDb (mxflg)) {
           rghtff=1;
           leftff=1;
           torght_B OR= rghtff;
           toleft_B OR= (leftff AND INV(mxsw));
25           }

```

If mxflg 62 is set and the rmxdrv_1 signal is asserted and valid, (acval), and logneg==1, indicating the most significant bit of the logic/shifter unit is 1, then the structure propagates 1's to the left and right PNs. torght_B and toleft_B are signals which propagate 1's through to the left or right along the arbitration bus. mxsw is used to disconnect one region from another so that local maximization functions may be performed.

35 The provision of the maximization architecture of the processor node array provides a structure in which all PNs in the array have general information about the state of other PNs in the array. As previously noted, one way to provide this information would be to charge a bus or line which connects to all PNs and let all of the PNs compare the value on the line or bus with the value contained in the PN data register. However, this requires an extraordinary amount of energy and, in a fault tolerant system, such as a neural network, electrical delays and other electrical considerations may affect

- 12 -

the integrity of the signal on such a wide spread bus. By providing the flag registers, or flip-flops, in each PN, data is shared amongst all of the PNs in the array through the use of very short connections, the right and left flip-flops acting as latches, where the output goes to a known state at a given time, thereby providing synchronous data to all of the processor nodes in the array.

In the event that there are more numbers to be maximized than there are processor nodes, a maximization can be performed upon several data figures contained in a single PN to determine which of the data figures has the largest value, and the maximization function amongst the PNs then can be run to determine which PN contains the maximum data figure.

The maximization architecture and function may also be used to determine a minimum data figure by providing an additional step of subtracting the segment with its values from 1, and then performing the previously described maximization function.

Although a preferred embodiment of the architecture and a preferred method of performing the maximization function has been described, it should be appreciated that variations and modifications may be made thereto without departing from the scope of the invention as defined in the appended claims.

Industrial Application

Processors constructed according to the invention are useful in neural network systems which may be used to simulate human brain functions in analysis and decision making applications, such as character recognition and robotic control.

- 13 -

WHAT I CLAIM IS:

1. A distributive digital maximization function architecture comprising:

an array (10) of processor nodes (28, 30, 32, 34), each node having manipulation units contained therein;

an input bus (36) connected to each node in said array;

an output bus (38) connected to each node in said array;

a data register (44) located in each processor node which receives a data figure from the manipulation unit, wherein said data figure consists of a plurality of segments, each segment having a value; and

maximization means located in each processor node, including - an arbitration bus (53) connected between adjacent processor nodes;

arbitration means (54), connected to said arbitration bus (53), for comparing a value of a segment to a value which is transmitted on said arbitration bus and for transmitting a comparison indicator;

a first indicator retainer means (46) and a second indicator retainer means (48), each connected to said arbitration means for retaining said comparison indicator; and

flag means (42) for flagging the processor node as containing a maximum data figure.

2. The architecture of claim 1 wherein said first indicator means and said second indicator means include, respectively, a left flip-flop (46) and a right flip-flop (48) located in each processor node for propagating said comparison indicator to all other processor nodes in the array.

3. The architecture of claim 2 wherein said arbitration bus (53) is connected between the right (48) and left (46) flip-flops in adjacent processor nodes.

- 14 -

4. The architecture of claim 2 wherein said arbitration means includes an OR gate (54) in each processor node having its inputs (56, 58) connected to said arbitration bus (53) and its output (60) connected to the right (48) and left (46) flip-flops in the particular processor node.

5. A distributive digital maximization function architecture, for use in an array (10) of processor nodes (28, 30, 32, 34), each node having an input bus (36), and output bus (38) and manipulation units therein, comprising, in a designated processor node (30):

a right flip-flop (48₁) and a left flip-flop (46₁),

an OR gate (54₁) having the inputs (56, 58) thereof connected to a right flip-flop (48₂) in the processor node (28) located immediately to the left of the designated processor node (30) and to the left flip-flop (46₂) in the processor node (32) located immediately to the right of the designated processor node (30), and having the output (60) thereof connected to the right flip-flop (48₁) and left flip-flop (46₁) in the designated processor node (30); and

a maximum value indicator (42₁).

- 15 -

6. A method of digitally determining a maximum data figure comprising:

providing an array (10) of processor nodes (28, 30, 32, 34) wherein each processor node contains -

a data register (44) for receiving a data figure, wherein the data figure includes a plurality of value-containing segments,

an arbitration structure for comparing a value of a segment to a signal provided by another processor node,

plural indicator retainers (46, 48) for retaining the signal, and

a flag register (42) for indicating that the processor node has a maximum data figure;

determining which processor node in the array has the largest data figure by -

setting the flag register (42) of each processor node with a positive flag;

examining the value of a subject segment in the data register of a designated processor node for a value of 1 or 0, and if and only if the value is 0, and the value of the subject segment of at least one other processor node is 1, setting the flag register for the designated processor node with a negative flag;

placing the value of the flag register (42) into the arbitration structure as a signal;

transmitting all signals contained in all arbitration structures to the indicator retainers (46, 48) regardless of the setting contained in the flag register (42); and

reiterating the steps beginning with examining until the processor node with the largest data figure is the only processor node having a positive flag.

1/3

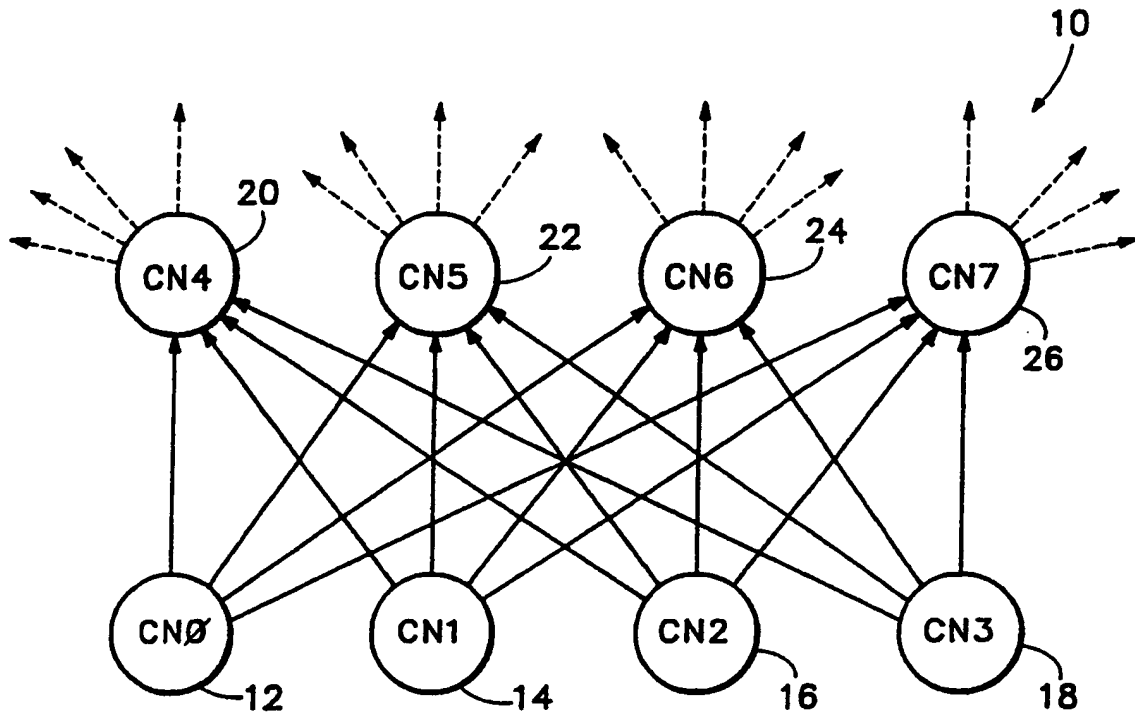


FIG. 1

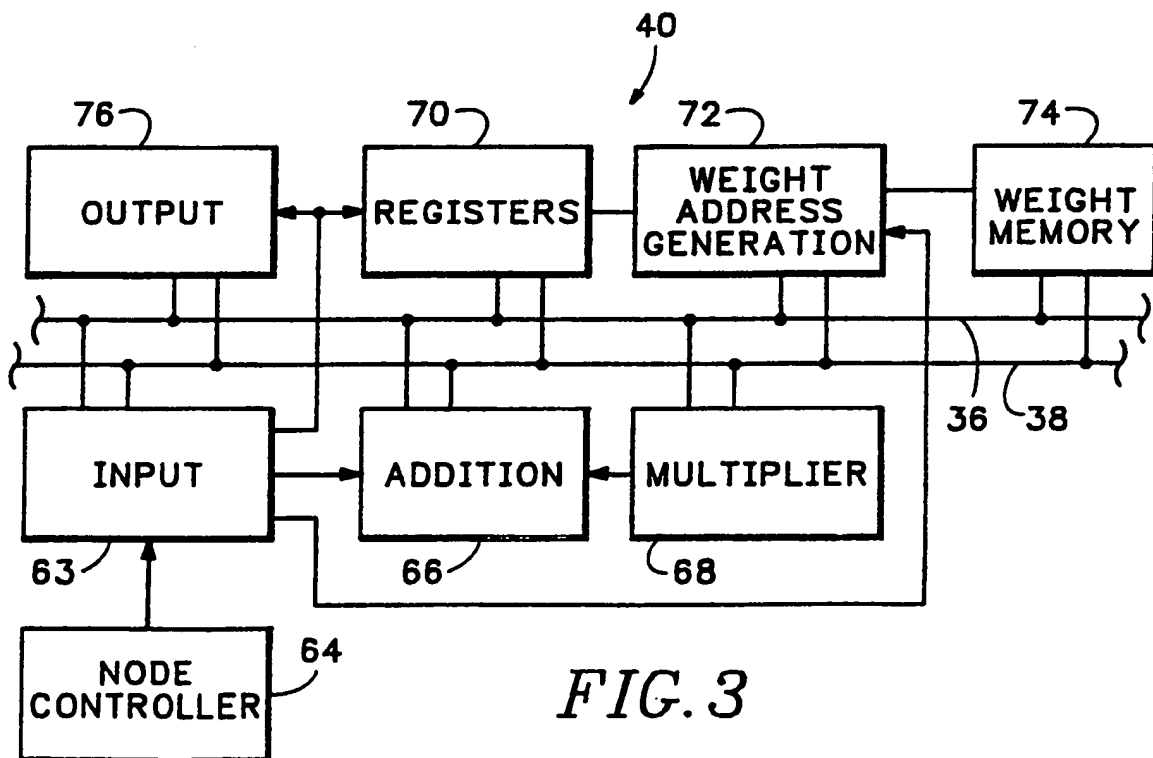


FIG. 3

2/3

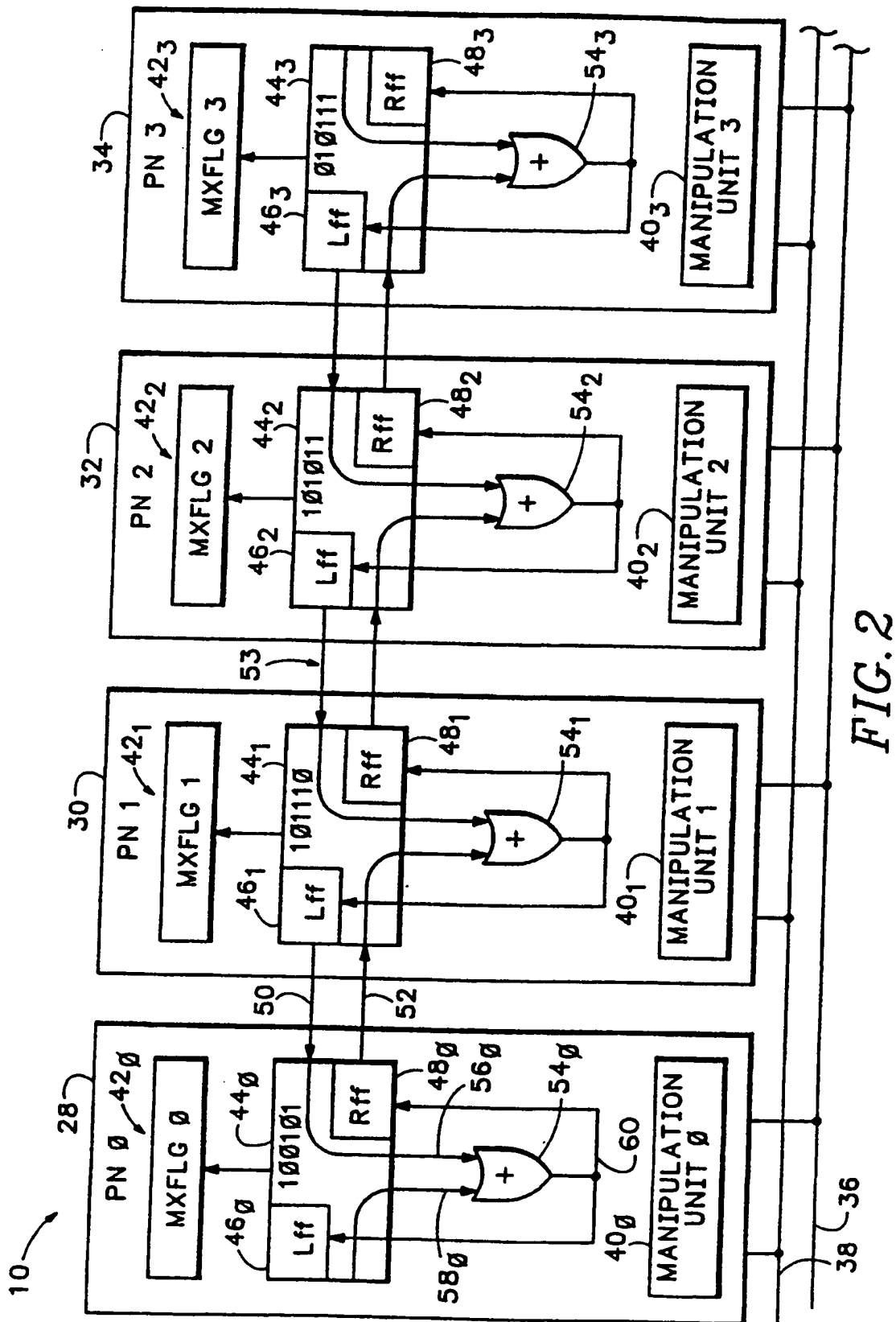
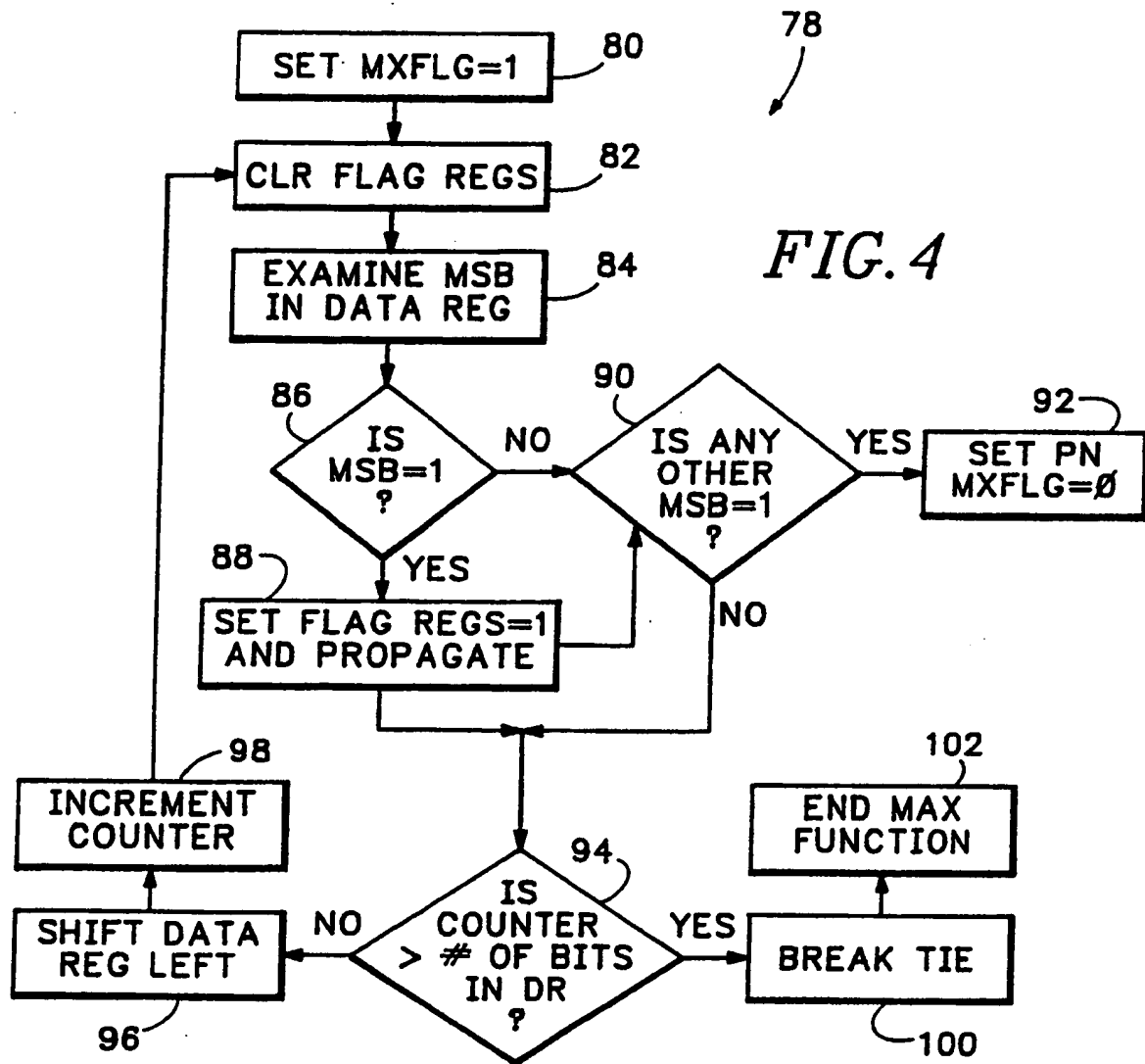


FIG. 2

3/3



INTERNATIONAL SEARCH REPORT

International Application No

PCT/US90/03068

I. CLASSIFICATION OF SUBJECT MATTER (If several classification symbols apply, indicate all) *		
According to International Patent Classification (IPC) or to both National Classification and IPC		
IPC (5)	:	G06F 15/18, 15/80, 7/02
U.S. Cl	:	364/200, 513
II. FIELDS SEARCHED		
Minimum Documentation Searched *		
Classification System		Classification Symbols
U.S.		364/200, 513, 900; 307/201
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched *		
III. DOCUMENTS CONSIDERED TO BE RELEVANT ^{1,2}		
Category *	Citation of Document, ^{1,6} with Indication, where appropriate, of the relevant passages ^{1,7}	Relevant to Claim No. ^{1,8}
A	US,A 3,484,749 (ANDREAE ET AL) 16 December 1969 See the entire document.	1-6
A	US,A 3,613,084 (ARMSTRONG) 12 October 1971 See the entire document.	
A	US,A 4,858,147 (CONWELL) 15 AUGUST 1989 See background of the invention.	
<p>* Special categories of cited documents: ^{1,3}</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"Δ" document member of the same patent family</p>		
IV. CERTIFICATION		
Date of the Actual Completion of the International Search ¹	Date of Mailing of this International Search Report ²	
25 SEPTEMBER 1990	15 FEB 1991	
International Searching Authority ¹	Signature of Authorized Officer ¹⁰ <i>Nguyen Ngoc Ho</i> NGUYEN NGOC-HO	
ISA/US	PAUL KULIK INTERNATIONAL DIVISION	